```c
#include <stdio.h>

// Function to swap two elements
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Partition function (Lomuto partition scheme)
int partition(int arr[], int low, int high) {
    int pivot = arr[high];  // Choose last element as pivot
    int i = (low - 1);

    for (int j = low; j < high; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }

    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}

// Quick Sort function (recursive)
void quickSort(int arr[], int low, int high) {
    if (low < high) {
```

```
        int pi = partition(arr, low, high);

        // Recursively sort elements before and after partition
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

// Function to print an array
void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main() {
    int n;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("\nArray before sorting:\n");
```

```
    printArray(arr, n);

    quickSort(arr, 0, n - 1);

    printf("\nArray after Quick Sort:\n");
    printArray(arr, n);

    return 0;
}
```

```
Enter number of elements: 5
Enter 5 elements:
2 5 2 9 6

Array before sorting:
2 5 2 9 6

Array after Quick Sort:
2 2 5 6 9


=== Code Execution Successful ===
```